



## Análisis de datos con Python 2024

### Tarea 02. Numpy

#### 1. Suma acumulativa (1D)

```
import numpy as np

# Crear un array unidimensional
arr1d = np.array([-1.5, 0.0, 2.3, -3.4, 4.6])

# Calcular la suma acumulativa
```

#### 2. Suma acumulativa (2D, eje 0)

```
# Crear un array bidimensional
arr2d = np.array([[ -1.5, 0.0, 2.3],
                  [ 4.6, -5.7, 6.8],
                  [-7.9, 8.0, -9.1]])

# Calcular la suma acumulativa a lo largo de las columnas
```

#### 3. Suma acumulativa (2D, eje 1)

```
# Crear un array bidimensional
arr2d = np.array([[ -1.5, 0.0, 2.3],
                  [ 4.6, -5.7, 6.8],
                  [-7.9, 8.0, -9.1]])

# Calcular la suma acumulativa a lo largo de las filas
```

#### 4. Media ignorando NaN (1D)

```
# Crear un array unidimensional con valores faltantes
arr1d = np.array([-1.5, 0.0, np.nan, 3.4, -4.6])

# Calcular la media ignorando NaN
```

5. Desviación estándar ignorando NaN (1D)

```
# Crear un array unidimensional con valores faltantes
arr1d = np.array([-1.5, 0.0, np.nan, 3.4, -4.6])

# Calcular la desviación estándar ignorando NaN
```

6. Mediana ignorando NaN (2D, eje 0)

```
# Crear un array bidimensional con valores faltantes
arr2d = np.array([[ -1.5, 0.0, 2.3],
                  [ 4.6, np.nan, -6.8],
                  [-7.9, 8.0, np.nan]])

# Calcular la mediana ignorando NaN a lo largo de las columnas
```

7. Varianza ignorando NaN (2D, eje 1)

```
# Crear un array bidimensional con valores faltantes
arr2d = np.array([[ -1.5, 0.0, 2.3],
                  [ 4.6, np.nan, -6.8],
                  [-7.9, 8.0, np.nan]])

# Calcular la varianza ignorando NaN a lo largo de las filas
```

8. Desviación estándar de un array 2D

```
# Crear un array bidimensional con valores faltantes
arr2d = np.array([[ -1.5, 0.0, np.nan],
                  [ 4.6, np.nan, -6.8],
                  [-7.9, 8.0, 9.1]])

# Calcular la desviación estándar ignorando NaN
```

9. Media a lo largo de un eje (2D)

```
# Crear un array bidimensional con valores faltantes
arr2d = np.array([[ -1.5, 0.0, np.nan],
                  [ 4.6, -5.7, 6.8],
                  [np.nan, 8.0, -9.1]])

# Calcular la media ignorando NaN a lo largo de las filas
```

10. Mediana de un array 1D

```

# Crear un array unidimensional
arr1d = np.array([-10.5, 0.0, 20.3, -30.7, 40.2])

# Calcular la mediana

```

11. Varianza de un array 1D

```

# Crear un array unidimensional
arr1d = np.array([-10.5, 0.0, 20.3, -30.7, 40.2])

# Calcular la varianza

```

12. Media de un array 2D a lo largo del eje 0

```

# Crear un array bidimensional
arr2d = np.array([[-1.5, 0.0, 2.3],
                  [4.6, -5.7, 6.8],
                  [0.0, 8.0, -9.1]])

# Calcular la media a lo largo de las columnas

```

13. Varianza de un array 2D a lo largo del eje 1

```

# Crear un array bidimensional
arr2d = np.array([[-1.5, 0.0, 2.3],
                  [4.6, -5.7, 6.8],
                  [0.0, 8.0, -9.1]])

# Calcular la varianza a lo largo de las filas

```

14. Desviación estándar a lo largo del eje 0

```

# Crear un array bidimensional
arr2d = np.array([[-1.5, 0.0, 2.3],
                  [4.6, -5.7, 6.8],
                  [0.0, 8.0, -9.1]])

# Calcular la desviación estándar a lo largo de las columnas

```

15. Media de un array 2D a lo largo del eje 1

```

# Crear un array bidimensional
arr2d = np.array([[-1.5, 0.0, 2.3],
                  [4.6, -5.7, 6.8],
                  [0.0, 8.0, -9.1]])

# Calcular la media a lo largo de las filas

```